

OLD-X 开源 飞控使用手册



云逸创新团队

您是否渴望拥有一个与 PX4 性能相当但在 KEIL 中使用 C 语言开发的飞控平台?

您是否渴望拥有一个不关心融合解算只需好好研究控制和导航理论的平台?

您是否对机器视觉, 感知障碍感兴趣却又不知如何为飞控加上 linux 与 ROS?

OLD-X 飞控力图帮您解决以上所有疑问!!!

更新日期: 2016/8/26 V1.0

免责声明.....	3
注意事项.....	3
QQ 讨论群: 567423074.....	3
1. OLD-X 飞控介绍.....	4
1.1 模块接口介绍.....	5
(1) IMU 模块.....	5
(2) FC 模块.....	7
(3) NRF-FC 模块.....	9
(4) POWER 模块.....	10
(5) NRF-G 模块.....	10
1.2 模块接线.....	11
2. OLD-X 使用入门.....	14
2.1 接收机连接方法.....	14
(1) 连接测试.....	14
(2) Keil 下检查通道映射关系.....	14
2.2 传感器校准方法.....	16
(1) 陀螺仪和加速度计.....	16
(2) 磁力计.....	16
2.3. 飞行器模式电调连接校准及转向确认.....	17
2.4. 波形显示与参数调节.....	18
(1) PC 上位机波形显示.....	18
(2) 调参方法.....	19
3. 飞控试飞流程.....	21
4. 注意事项.....	22
4.1 定高.....	22
(1) 模式切换.....	22
(2) 注意事项.....	23
4.2 定点模式.....	23
(1) 模式说明.....	23
(2) 注意事项.....	23
5. 进一步开发说明.....	24

免责声明

OLD-X 飞控为开源产品因此在下列原因不在免费保修范围：

1. 在室外极限环境下使用如大风，大雨等；
2. 用户主观操作失误造成的损失；
3. 用户自行修改飞控源码后造成的事故损失；

OLD-X 飞控为开源产品无法保证其与商业飞控一样具有高稳定性和可靠性，在飞行器工作时可能会发生飞行器坠落以及对他人伤害的可能因此在如下情况下团队不承担相关责任：

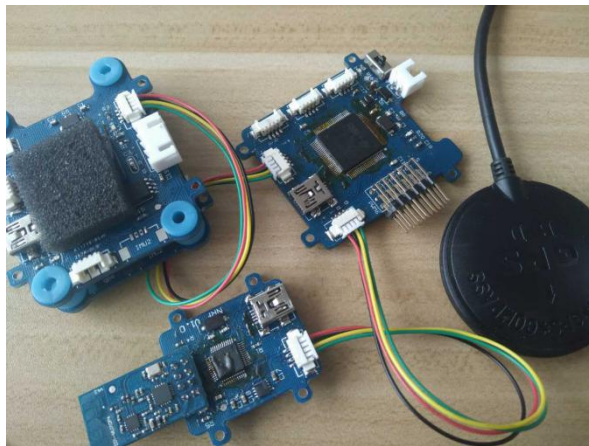
1. 用户主动或故意操控飞行器造成人身伤害和财产损失；
2. 用户自行修改飞控源码后造成的事故损失；
3. 用户在低电量情况下执意飞行造成的事故损失；
4. 用户在强风，强磁力干扰，高速飞行情况下造成的事故损失；

注意事项

- 1、 安装飞行器时，请确保飞行器重心在机架中心，有负载的在机架中心的垂直方向上。
- 2、 安装主控制器时，尽量安装在靠近中心位置，确保主控印有标记的一面朝上，并使其与机身水平面保持平行，否则会导致飞行器水平方向飘移。
- 3、 主控制器安装有方向要求，务必使箭头的朝向与飞行器机头方向一致。
- 4、 在固件升级、调试过程中请断开电调与电池的连接或移除所有桨翼！
- 5、 飞行时切记先打开遥控器，然后启动多旋翼飞行器！着陆后先关闭飞行器，再关闭遥控器！
- 6、 低压情况下请及时降落
- 7、 光流定点请在明亮光照情况下使用
- 8、 光流制动适用于低速航线情况，高速情况可能出现刹车失效请格外注意

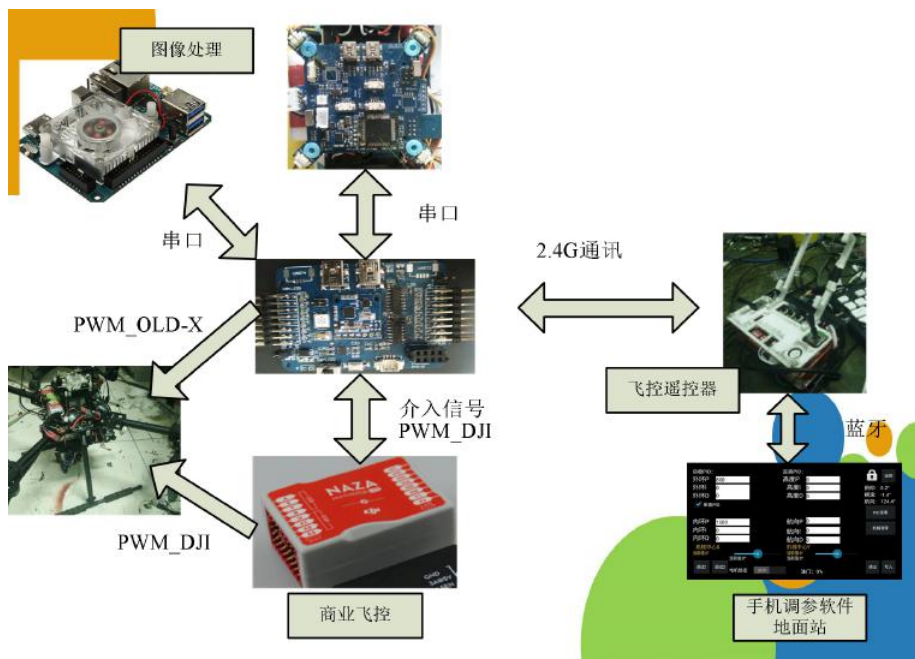
QQ 讨论群： 567423074

1. OLD-X 飞控介绍



OLD-X 飞控是国内开源飞控中为数不多基于嵌入式操作系统的开源飞控，目前国内的开源飞控层次不穷但大多都是互相抄来抄去，只是在硬件上做出稍微的改变，在飞控软件架构和控制方法上并没有本质的区别，用户购买往往只能是重复投资无法学习到新的控制理论知识和嵌入式编程技巧。

OLD-X 飞控基于 UCOSII 操作系统大大增加了对 CPU 的利用率比起其他裸奔的开源飞控代码执行效率和执行频率稳定性高出很多，除了采用操作系统外 OLD-X 飞控并没有采用纯串级 PID 的控制方式而是采用了自抗扰的控制方法，该方法对飞行器外部扰动和为建模部分的估计比传统 PID 积分有更好的控制效果，同时有保留了像 PID 一样易于调参的特性。



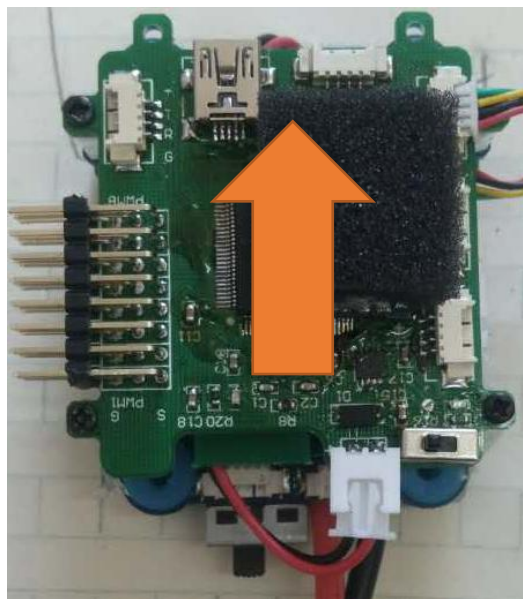
评价一款开源飞控是否成熟的标准是是否具有定位和可靠的定高能力，OLD-X 飞控通过 UKF 算法完成加速度计和光流模块的融合实现室内外稳定的悬停定位，在目前的测试中 OLD-X 飞控能有效抵御 3 级的风力稳定悬停，在定高方面 OLD-X 飞控基于 EKF 融合气压计与加速度计，而市面上其他开源飞控采用互补滤

波融，OLD-X 飞控具有更好悬停定高和高速航线飞行性能，在高速侧飞时 OLD-X 飞控吊高不明显在高速刹车制动时也不会突然吊高而这是其他开源飞控难以媲美的。

OLD-X 飞控采用模块化设计 IMU，飞控，数传划分清晰不像其他飞控在更新硬件时需要整体升级价格过高，同时通过不同价格等级的模块可以满足不同用户的需求，同时 OLD-X 飞控另一大特点是 SWD 配套 USB 转接模块进行程序下载，不同与其他开源飞控下载和调试程序时要繁琐地查看引脚顺序 OLD-X 飞控只需要插上 USB 既可以完成程序的下载同时具有良好的抗干扰性能，未来团队还将更新 GPS 导航及视觉导航等功能请大家期待。

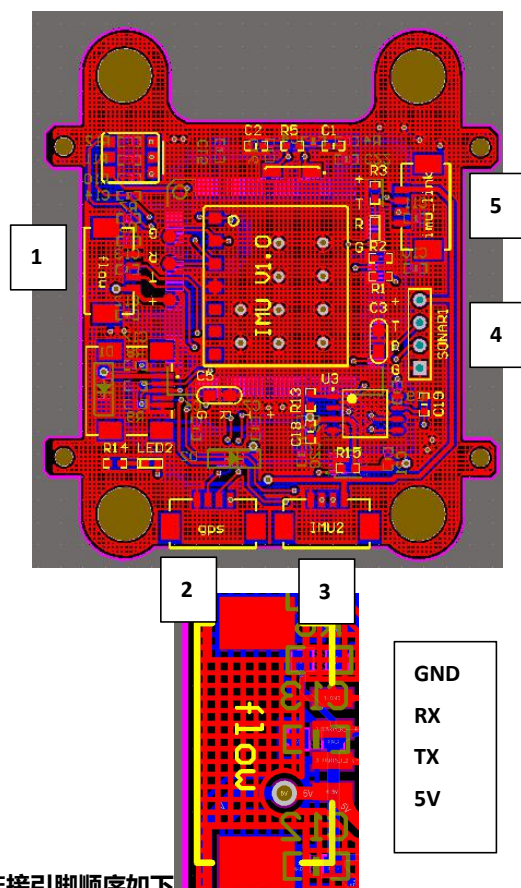
1.1 模块接口介绍

OLD-X 飞控为塔式一体化集成设计，包括 FC 模块，IMU 模块，NRF 模块，POWER 模块，机头方向为箭头方向在安装时请问出错：

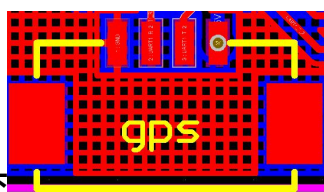


(1) IMU 模块

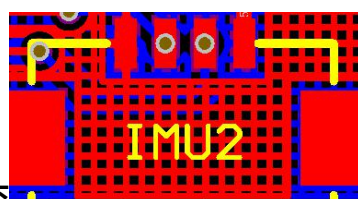
IMU 模块接口如下：



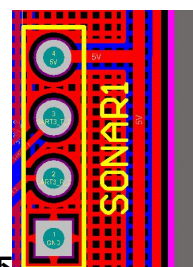
1:为 flow 接口与光流模块连接引脚顺序如下




2:为 GPS 接口与 GPS 模块连接引脚顺序如下



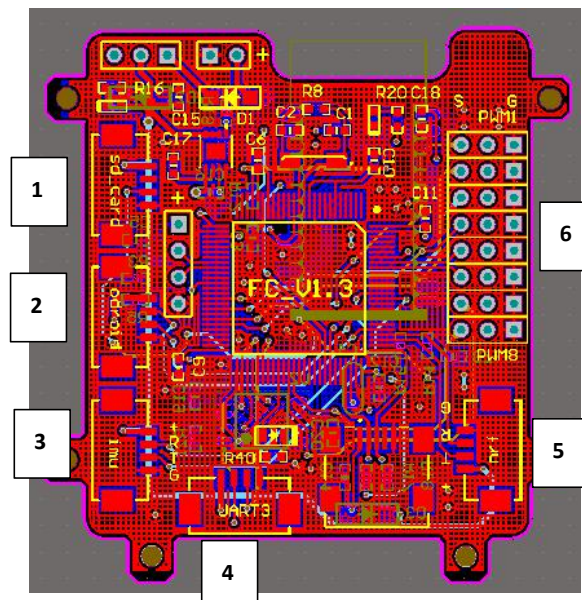
3:为备用接口与 IMU2 或壁障模块连接引脚顺序如下

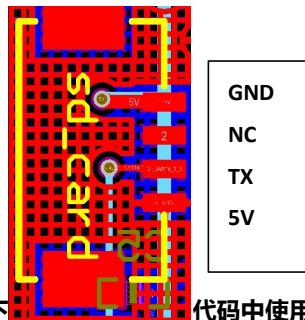


4:为 sonar 接口与超声波模块连接（默认使用 US-100 串口模式）引脚顺序如下

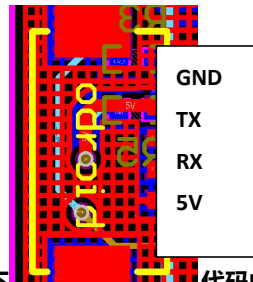


FC 模块接口如下：

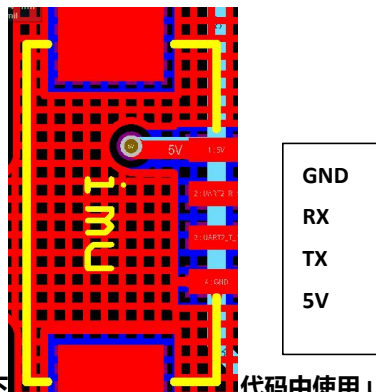




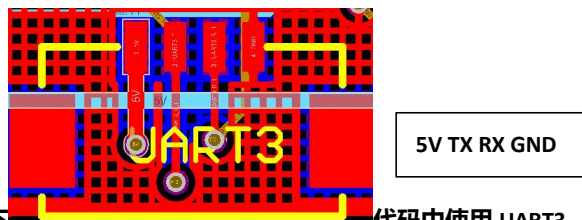
1:为 SD 接口与 SD 存储模块连接引脚顺序如下 代码中使用 UART4



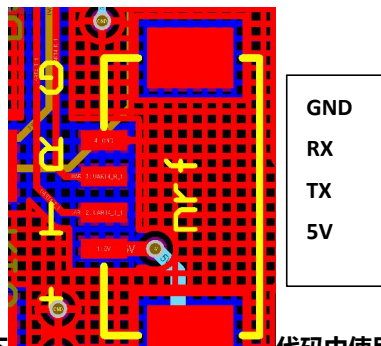
2:为图像接口与 Odroid 或树莓派连接引脚顺序如下 代码中使用 UART5



3:为 Gol-link 接口与 IMU 模块连接引脚顺序如下 代码中使用 UART2

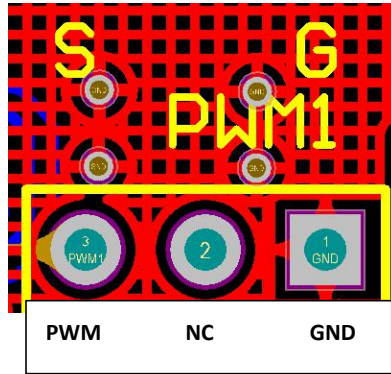


4:为备用接口引脚顺序如下 代码中使用 UART3



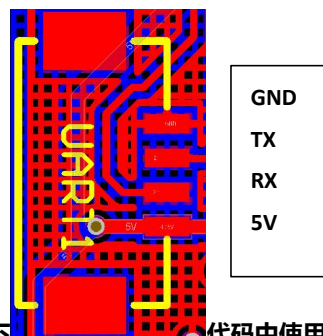
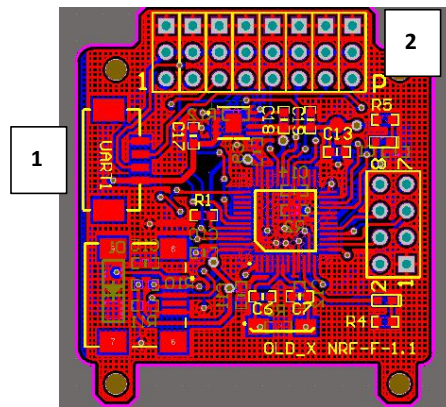
5:为 nrf 接口与 NRF 模块连接引脚顺序如下 代码中使用 UART5

6:为 PWM 输出接口从上到下为 PWM1~8 接口顺序如下：



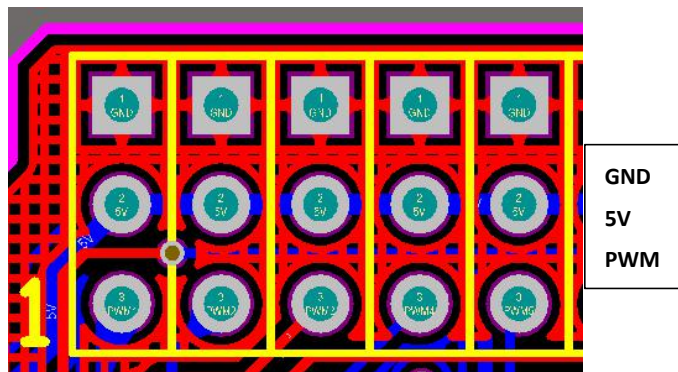
(3) NRF-FC 模块

NRF-FC 模块采集了遥控接收机信号同时也接受地面站数据接口如下：

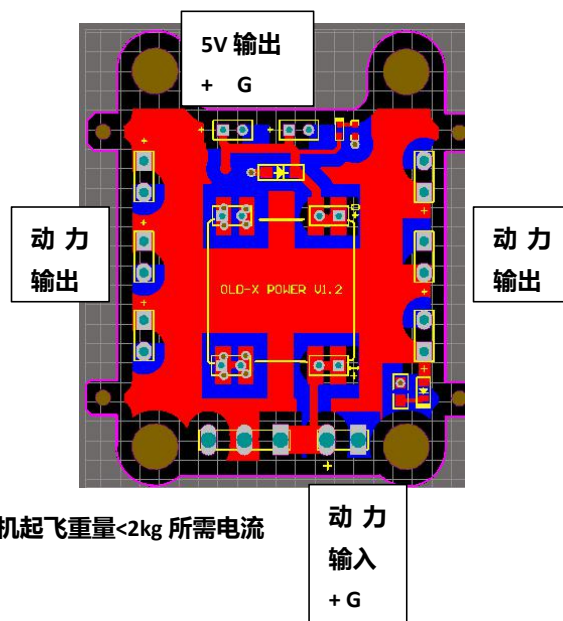


1:为 NRF 接口与 FC 模块连接引脚顺序如下 代码中使用 UART1

2:为 PWM 输入接口与接收机连接从做到右对应 PWM1~8



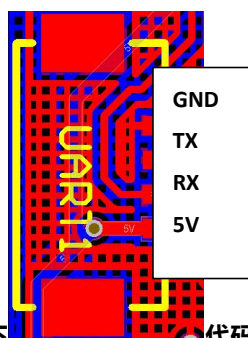
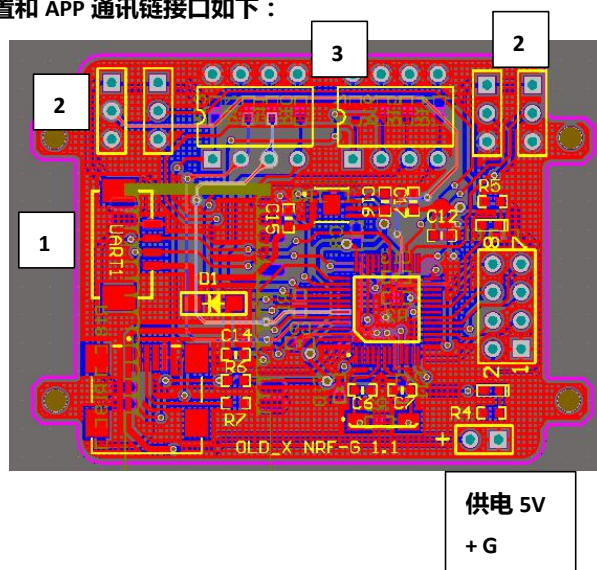
(4) POWER 模块



最大输入支持 4S 电池，飞机起飞重量 < 2kg 所需电流

(5) NRF-G 模块

NRF-G 主要完成模式设置和 APP 通讯链接口如下：



1:为 PC 接口与 PC 机连接模块连接引脚顺序如下

代码中使用 UART1，为 TTL 电平需要

USB-TTL 模块才能与 PC 机连接

2：为 KEY_SEL 从左到右对应代码中 0~3，如：

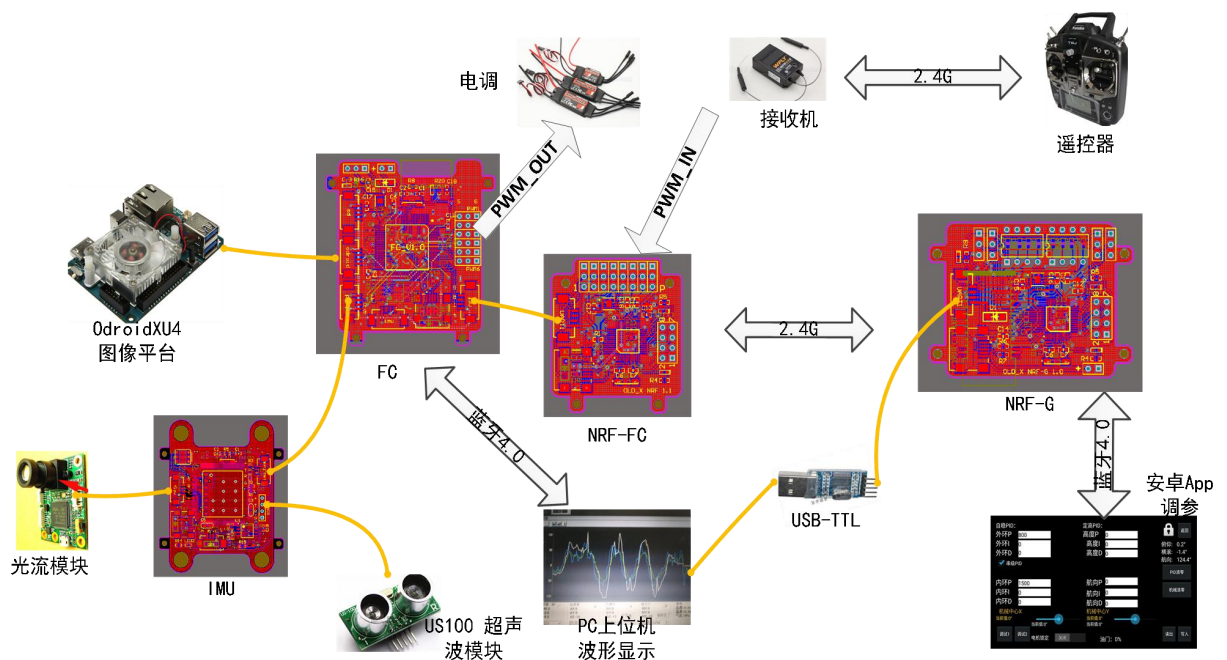
```
mode.en_sd_save=KEY_SEL[1];  
mode.en_pid_sb_set=KEY_SEL[2]; //使能PID设置
```

3：为 KEY 从左到右对应代码中 0~7，如：

```
mode.att_pid_tune=KEY[6]&&KEY[5]&&KEY[3]&&KEY[2]&&KEY[1]&&KEY[0];
```

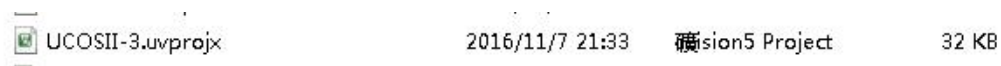
注：该模块功能只有购买的用户才能使用

1.2 模块接线



1.3 KEIL5 安装使用和程序下载

打开工程已 FC 源码为例如果 KEIL5 安装好后再 FC 源码文件\OLD_X_FC\USER 目录下点击

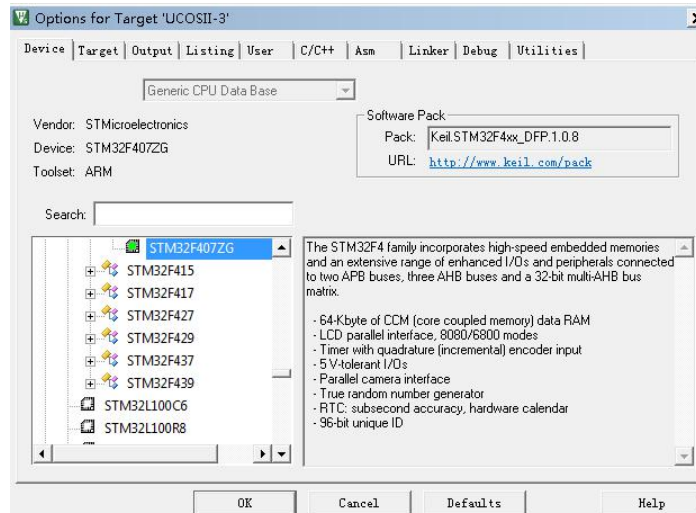


KEIL5 应当会自动打开该工程。

OLD-X 飞控开发默认使用 KEIL5，在使用时你可能会遇到如下问题：

(1) KEIL5 安装：KEIL5 安装请自行百度安装破解方法，资料中也提供了安装文件

(2) F4 库安装：安装好 KEIL5 后并不能直接编译我们的工程，你还需要安装 F4 的库，具体可以参考正点原子 F4 的板子，如果正常后则 KEIL 设置中应该出现我 F4 芯片的型号：

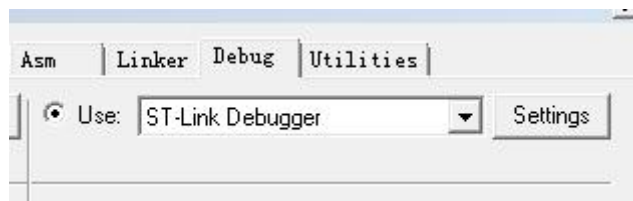


点击  中间那个增量编译如果出现：

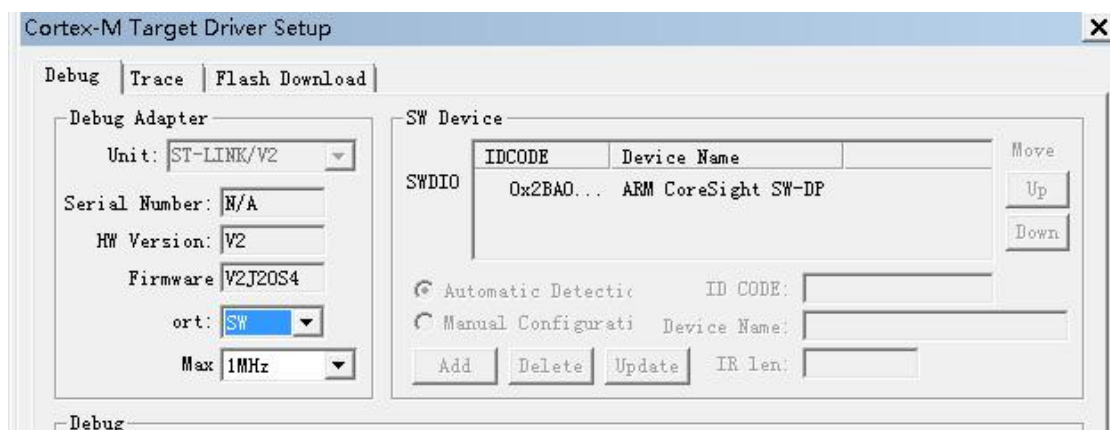
```
Build target 'UCOSII-3'
linking...
Program Size: Code=183686 RO-data=3818 RW-data=4508 ZI-data=59964
FromELF: creating hex file...
"...\\OBJ\\UCOSII-3.axf" - 0 Error(s), 0 Warning(s).
```

则表示配置成功编译没错。

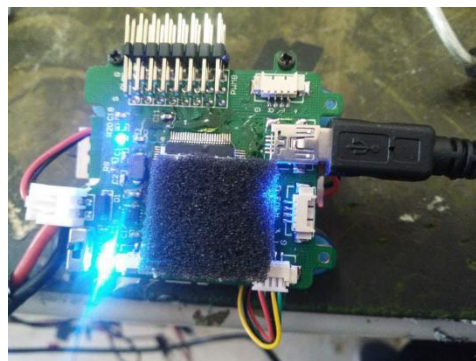
(3) 选择 SWD 模式下载：很多人在第一次使用 STM32 下载程序时都会因为选择下载方式错误而导致程序无法下载，默认 KEIL5 是使用 j-link 作为下载器，我们使用的是 SWD 下载方式设置在



点击 Setting 连接上板子 SWD 供电应该能看到显示芯片的型号：

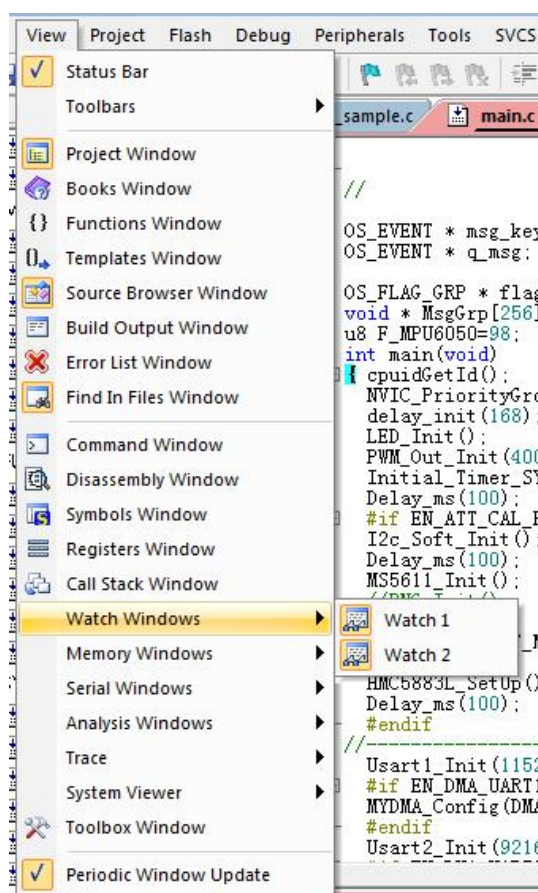




(4) 使用 SWD 转接板进行下载：为方便调试 OLD-X 飞控将 SWD 接口做成了 USB 接口的形式因此如果不使用转接板是无法完成代码的下载的，转接板上已经标注了对应下载器需要的连线顺序使用杜邦线与下载器连接好在通过一根 USB 线连接电路板即可，注意：调试时 5V 信号不要接，电路默认供电采用电池降压 5V 供电，使用电脑 USB 供电功耗无法满足飞控会导致程序无法下载或者程序跑飞。



(5) 下载程序和 Debug：点击  可以下载程序，重新上电或者点击  后程序才会运行；

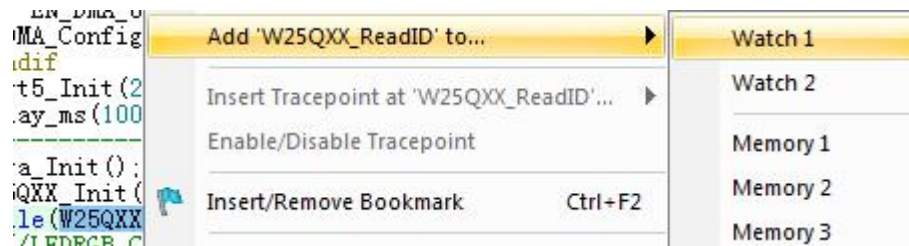
点击  也会下载程序并自动进入调试模式，在调试模式需要注意的是首先选择



添加 watch，同时勾选最后一项使 watch 中的数据会实时刷新，点击  程序才会运行，点击  会推出 debug 模式，在该模式下单击代码灰色部分会添加断点：

```
58 Initial_Timer_SYS();
59 Delay_ms(100);
60 #if EN_ATT_CAL_F
```

但是 keil 调试器本身单步调试有问题，想单步调试只能手动添加多个连续的断点，在代码中选择任意变量鼠标右键选择



能将其添加到 watch 中。

其他 STM32 相关学习资料已经 UcosII 系统开发方法可以参考正点原子 F4 和附带的资料。

2. OLD-X 使用入门

2.1 接收机连接方法

(1) 连接测试

接收机 NRF-G 模块通道对应遥控器接收机通道映射关系为 NRF-PWM 输入通道：

通道 1	横滚通道
通道 2	俯仰通道
通道 3	油门通道
通道 4	偏航通道
通道 5	定高模式通道
通道 6	NC
通道 7	手动，定点模式切换通道
通道 8	飞控解锁通道

(2) Keil 下检查通道映射关系

A.建议在第一次使用接收机或者更换接收机后进行通道映射检查，打开 NRF-G Keil 下工程，连接调试器，点击 Debug 按钮，点击 run，在 usart_fc.c 文件中找到 Rc_Get_Pwm 右击选择加入 watch1，点击+号查看详情，

```
1130 Rc_Get_PWM.PITCH=((int16_t) (*(data_buf+4)<<8) | *(data_buf+5));
1131 Rc_Get_PWM.ROLL=((int16_t) (*(data_buf+6)<<8) | *(data_buf+7));
1132 Rc_Get_PWM.THROTTLE=((int16_t) (*(data_buf+8)<<8) | *(data_buf+9));
1133 Rc_Get_PWM.YAW=((int16_t) (*(data_buf+10)<<8) | *(data_buf+11));
1134
1135 Rc_Get_PWM.AUX1=((int16_t) (*(data_buf+12)<<8) | *(data_buf+13));
1136 Rc_Get_PWM.RST=((int16_t) (*(data_buf+14)<<8) | *(data_buf+15));
1137 Rc_Get_PWM.HEIGHT_MODE=((int16_t) (*(data_buf+16)<<8) | *(data_buf+17));
1138 Rc_Get_PWM.POS_MODE=((int16_t) (*(data_buf+18)<<8) | *(data_buf+19));
```

首先拨动遥控器俯仰观察 Pitch 数据是否变化，值是否在 1000~2000 内 且在不动时大幅度突变，遥感中位为 1520，用同样方法测试横滚，油门和航向通道，如果都正常则进行下一步。

B.检测模式开关

由于每个遥控器开关对应的遥感量不一样因此有可能造成模式切换无法对所有遥控器都适用，因此在使用时应该首先检查各模式开关，如果对应遥控量无法达到判断条件可以通过配置遥控器修改当然也可以通过

直接修改代码这样更加便捷。

首先是定高模式，定高模式分为手动（油门遥感直接对应油门输出），气压定高以及超声波定高。模式默认选用了遥控器一个旋钮作为模式切换条件代码中 Rc_Get_PWM.HEIGHT_MODE 为输入的遥感量可以拨动旋钮观察是否变化，同时 RX_CH_PWM[AUX4r]=Rc_Get_PWM.HEIGHT_MODE 定高模式判断条件在 rc.c 中：

```
286     if( RX_CH_PWM[AUX4r] >1900 )
287     {
288         if(ultra_ok == 1)
289         {
290             height_ctrl_mode = 2;//超声波
291         }
292         else
293         {
294             height_ctrl_mode = 1;
295         }
296     }
297     else if(RX_CH_PWM[AUX4r] >1400 &&RX_CH_PWM[AUX4r] <1600 )
298     {
299         height_ctrl_mode = 1;//气压计
300     }
301     else if(RX_CH_PWM[AUX4r] <1200 )
302     {
303         height_ctrl_mode = 0;
304     }
305 }
306
307
```

可以修改 1900,1400,1200 等值改变判断调节，修改完成后建议添加 height_ctrl_mode 到 watch 中观察是否按照想要的条件变化。

c.检查定点模式 开关

定点模式默认使用了一个三档开关，分别对应手动自稳，光流定点以及智能控制模式（图像导航，轨迹飞行模式），Rc_Get_PWM.POS_MODE 为遥控量输入同时 RX_CH_PWM[AUX3r]=Rc_Get_PWM.POS_MODE，模式切换条件在 rc.c 中

```
308
> 309     if(RX_CH_PWM[AUX3r]>1800)
310         mode.flow_hold_position=2;
311     else if(RX_CH_PWM[AUX3r]<1400)
312         mode.flow_hold_position=0;
313     else
314         mode.flow_hold_position=1;
315
```

可以修改 1800,1400 对应不同遥控器和开关模式，目前版本暂时未包括智能控制因此 mode.flow_hold_position==1 表示使用光流定点模式。

D.电机解锁判断

注意电机解锁是否重要由于 OLD-X 为开源飞控不能保证在飞行中不会应为 bug 导致炸鸡因此为保护人员安全采用开关进行电机解锁而不是像 NAZA 等 DJI 飞控掰遥感的方式。模式默认选用两档开关切换条件在 ucos.c 中：

```
232     #if 1  FUSE_RC_GROUND
233     if(Rc_Get_PWM.AUX1>1500)
234         fly_ready=1;//mode.en_sd_save=1;
235     else
236         fly_ready=0;// mode.en_sd_save=0;
237     #endif
```

Fly_ready 为 1 表示电机以及解锁可以起飞。

2.2 传感器校准方法

(1) 陀螺仪和加速度计

建议在第一次使用 OLD-X 飞控时进行陀螺仪加速度计的校准，打开 FC 文件 Keil 下的工程，FC 版连接调试器，点击 Debug 按钮，点击 run，在 mpu6050.c 中找到

```
6
7 MPU6050_STRUCT mpu6050_fc, mpu6050;
8
```

两个结构体 mpu6050_fc 为 FC 模块的传感器，另一个为 IMU 模块中的传感器，右击选择将两个结构体都加入 watch1，点击+号查看详情，两个结构体校准方法都一样：

A.将飞控或飞行器水平放置

B.将 MPU6050.Acc_CALIBRATE 设为 1 校准加速度计，但该数自动回到 0 时校准完毕，注意只校准加速度不会将校准值写入 flash 中。

C.将 MPU6050.Gyro_CALIBRATE 设为 1 校准加速度计，但该数自动回到 0 时校准完毕，同时将当前的加速度计和陀螺仪校准值写入 flash 中

D.点击查看新的校准值 off_set 项

E.点击 RST 复位芯片重新运行查看偏差值是否写入 FLASH 中

F.用同样方法校准 MPU6050_fc 结构体

Name	Value	Type
mpu6050	0x2000B374 &mpu6050	struct <untag...
Acc_CALIB...	0	char
Gyro_CALI...	0	char

mpu6050_fc	0x2000B2F8 &mpu605...	struct <
Acc_CALIBRATE	0	char
Gyro_CALI...	0	char

(2) 磁力计

建议在第一次使用 OLD-X 飞控时或大区域更换使用时进行磁力计的校准，打开 FC 文件 Keil 下的工程，FC 版连接调试器，点击 Debug 按钮，点击 run，在 hml_sample.c 文件下找到：

```
0 ak8975_t ak8975 = { {0, 0, 0}, {124, -449, 369}, {1, 0.532, 0.486}, {0, 0, 0} };
1 ak8975_t ak8975_fc = { {0, 0, 0}, {124, -449, 369}, {1, 0.532, 0.486}, {0, 0, 0} };
```

右击选择加入 watch1，其中_fc 为 FC 模块中的磁力计另一个为 IMU 模块的磁力计，目前版本没有采集 GPS 数据因此不需要对 IMU 的磁力计进行校准，点击+号查看详情，

G.将飞控或飞行器水平放置

H.将 ak8975_fc.Mag_CALIBRATED 设为 1

I.以 8 字形旋转模块，当该数自动回到 0 时表示校准完毕

J.点击 Off_set 和 Gain 查看新的校准值

K.点击 RST 复位芯片重新运行查看偏差值是否写入 FLASH 中

L.水平放置模块随意转动 90°航向与实际地理信息对比是否准确快速，同时是否有飘逸否则重新校准

用户可以自行编程设置开关对 ak8975_fc.Mag_CALIBRATED 赋值实现遥控器控制校准。

ak8975_fc	0x200003E8 &ak8975_fc	struct <untag
Mag_Adc	0x200003E8 &ak8975_fc	struct <untag
Mag_Offset	0x200003F0	struct <untag
Mag_Gain	0x200003FC	struct <untag
Mag_Val	0x20000408	struct <untag
Mag_CALI...	0	unsigned char

2.3.飞行器模式电调连接校准及转向确认

飞控目前版本只适用于四轴飞行器如果想控制其他类型飞行器用户可自行移植其他飞控源码对 att_control.c 文件中：

```

824 posture_value[0] = - out_roll + out_pitch + out_yaw ;
825 posture_value[1] = + out_roll + out_pitch - out_yaw ;
826 posture_value[2] = + out_roll - out_pitch + out_yaw ;
827 posture_value[3] = - out_roll - out_pitch - out_yaw ;

```

进行修改，飞控 FC 模块 PWM 接口与电调信号线连接，PWM1~4 对应电调电机 M1~M4 对应连接顺序和电机转向如下图所示：



X型四旋翼飞行器

第一次飞行时推荐对电调进行校准，校准时请不要安装螺旋桨以免造成伤害，我们对此不负责，关闭动力电源，通过 ST-link 连接飞控打开工程介入 Debug 模式在 pwm_out.c 文件中

```

146 TIM_Cmd(TIM4, ENABLE);
147
148 if(motor_cal)
149 MOTOR_SET();
150 TIM1->CCR1 = PWM_RADIO *( 0 ) + INIT_DUTY; //5
151 TIM1->CCR2 = PWM_RADIO *( 0 ) + INIT_DUTY; //6
152 TIM1->CCR3 = PWM_RADIO *( 0 ) + INIT_DUTY; //7
153 TIM1->CCR4 = PWM_RADIO *( 0 ) + INIT_DUTY; //8
154 TIM4->CCR1 = PWM_RADIO *( 0 ) + INIT_DUTY; //7
155 TIM4->CCR2 = PWM_RADIO *( 0 ) + INIT_DUTY; //8
156

```

和

```

13 void MOTOR_SET(void)
14 {
15
16 Delay_ms(4000);
17 TIM1->CCR1 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //5
18 TIM1->CCR2 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //6
19 TIM1->CCR3 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //7
20 TIM1->CCR4 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //8
21 TIM4->CCR1 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //7
22 TIM4->CCR2 = PWM_RADIO * ( 1000 ) + INIT_DUTY; //8
23 Delay_ms(4000);
24 TIM1->CCR1 = PWM_RADIO * ( 0 ) + INIT_DUTY; //5
25 TIM1->CCR2 = PWM_RADIO * ( 0 ) + INIT_DUTY; //6
26 TIM1->CCR3 = PWM_RADIO * ( 0 ) + INIT_DUTY; //7
27 TIM1->CCR4 = PWM_RADIO * ( 0 ) + INIT_DUTY; //8
28 TIM4->CCR1 = PWM_RADIO * ( 0 ) + INIT_DUTY; //7
29 TIM4->CCR2 = PWM_RADIO * ( 0 ) + INIT_DUTY; //8
30
31 }

```

处添加两个断点将 motor_cal 添加进 watch 中点击 run，程序会停在第一个断点处此时在 watch 中将 motor_cal 设为 1 后再次点击 run 之后跑到第二个断点后接通动力电源，一般电调在此时会给出提示音具体参考购买的电调说明书，之后再次 run 电调应当会长响之后完成校准具体提示音也请参考购买电调说明书。

2.4.波形显示与参数调节

(1) PC 上位机波形显示

PC 端波形显示用到的软件为第七实验室推出的上位机，使用方法是首先在 PC 端插入蓝牙主机模块，然后等待其与 FC 板的配对，如果配对成功则 FC 板上 LED 会常亮，此时打开数据接收则可以在上位机中实时显示波形，由于为了更高的通讯速率以保证波形实时性因此 OLD-X 使用的是蓝牙 2.0 模块，通讯距离在 10M 内可以满足在室内实验波形显示的需求，如要远程波形显示可以使用 NRF-G 模块的数传功能。



选择发送数据内容在 ucos.c 中串口线程里通过 DMA 发送数据：

```
329 //=====串口 任务函数=====
330 OS_STK UART_TASK_STK[UART_STK_SIZE];
331 u8 UART_UP_LOAD_SEL=0; //-----UART UPLOAD DATA SEL
```

默认提供了十多个常用调试数据发送例子，用户可自行在：

```
537 (int16_t)(Yaw*10), (int16_t)(Pitch*10.0), (int16_t)(Roll*10.0), thr_test, 0, 0/10, 0)
538 case 24://TRIG PID TUNNING
539 data_per_uart1(
540 0, 0, d_flow_watch[0]*100,
541 0, 0, d_flow_watch[1]*100,
542 0, acc_body[1]*100, 0,
543 (int16_t)(Yaw*10), (int16_t)(Pitch*10.0), (int16_t)(Roll*10.0), thr_test, 0, 0/10, 0)
544
```

之后按照之前的例子添加所需发送的数据，主要 case 语言后需要有对应的 break。

(2) 调参方法

A.未购买了 NRF-G 的用户

购买了 NRF-G 的用户默认情况下是使用的，因此未购买 NRF-G 模块的用户暂时不支持安卓 APP 的远程调参功能用户可以参考 NRF-G 代码自行移植使用 FC 上的蓝牙模块进行调参，未购买 NRF-G 的用户默认使用调整 PID 参数的方式实现参数调节，通过修改 parameter.c 文件中：

```
331 pid_setup.groups.ctrl2.roll.kp=pid_setup.groups.ctrl2.pitch.kp =0.425;//0.66;//0.35;//28;/
332 SPID.IP=pid_setup.groups.ctrl2.roll.kp*1000;
333 //i
334 pid_setup.groups.ctrl2.roll.ki =pid_setup.groups.ctrl2.pitch.ki = 0.00;//5;//5;//0.15;//0.
335 SPID.II=pid_setup.groups.ctrl2.roll.ki*1000;
336 //d
337 pid_setup.groups.ctrl2.roll.kd = pid_setup.groups.ctrl2.pitch.kd =0.5;//0.3;//0.3;//2;//0.
338 SPID.ID=pid_setup.groups.ctrl2.roll.kd*1000;
339 //-----GRO-----
340 //p
341 pid_setup.groups.ctrl1.roll.kp=pid_setup.groups.ctrl1.pitch.kp = 0.425;//0.466;//0.3;//0.4
342 SPID.OP=pid_setup.groups.ctrl1.roll.kp*1000;
343 //i
344 pid_setup.groups.ctrl1.roll.ki =pid_setup.groups.ctrl1.pitch.ki =0.1;// 0.05;//0.1;
345 SPID.OI=pid_setup.groups.ctrl1.roll.ki*1000;
346 //d
347 pid_setup.groups.ctrl1.roll.kd=pid_setup.groups.ctrl1.pitch.kd =8;// 2.0;//4;//2.0;
348 SPID.OD=pid_setup.groups.ctrl1.roll.kd*1000;
349 //-----YAW-----
350 pid_setup.groups.ctrl2.yaw.kp = 0.8;//0.4;//0.8;
351 SPID.YD=pid_setup.groups.ctrl2.yaw.kp*1000;
352 pid_setup.groups.ctrl2.yaw.ki = 0.05;
353 SPID.YI=pid_setup.groups.ctrl2.yaw.ki*1000;
354 pid_setup.groups.ctrl2.yaw.kd = 0.3;//0.1;//0.3;
355 //-----
356 pid_setup.groups.ctrl1.yaw.kp = 0.8;//1.2;//0.8;
357 SPID.YP=pid_setup.groups.ctrl1.yaw.kp*1000;
```

参数实现姿态 PID 和定点 PID 的条件，通过修改 height_ctrl.c 文件中：

```
82 default:
83 ultra_pid.kp = 0.88;
84 ultra_pid.ki = 0.05;
85 ultra_pid.kd = 1.2;
86 break;
87 }

119 default:
120 wz_speed_pid.kp = 0.35;//1.25;//0.3;//0.25;/
121 wz_speed_pid.ki = 0.1;//225;//1.4;//0.08;//0
122 wz_speed_pid.kd = 2.0;//0.5;//8;//1.5;
123 break;
```

处参数调节定高 PID，烧写程序后重新试飞判断是否调节好飞机状态，该方法是否麻烦因此推荐购买 NRF-G 模块，注意烧写时请保证断开了飞行器动力电源同时点击上锁。

B.购买 NRF-G 的用户

购买了 NRF-G 模块的用户可以通过手机 APP 实现参数调节，第一次使用关闭 FC 模块的电源后打开 NRF-G 模块电源打开 APP 点击蓝牙连接，搜索周边蓝牙模块，成果后第二次使用会默认显示不需要再次搜索。



点击飞行控制后进入主界面此时打开 FC 模块电源后应当能看到 FC 的姿态角等消息在屏幕上显示



将会变成蓝牙标志。



点击 PID 设置进入远程调参界面，由于软件 BUG 目前高度 PID 无法使用请不要尝试。



通过电机和输入能改变各参数值通过电机写入能将参数发送到飞控中，但是注意当 NRF-G 模块第三个两档开关拨上去时参数才被真正写入否则数据不好存储下来，可以电机读出查看写入的参数。发送到飞控的参数为 9 个赋给了 SPID.OP OI OD IP II ID YP YI YD 这几个数通过修改可以对任意参数进行设置，在 rc_mine.c 文件中：


```

276 // else if(KEY[0]==1&&KEY[1]==1){
277 //     pid.avoid.out.p=0.001*(float)SPID.OP;
278 //     // if(!mode.height_safe){
279 //         ultra_pid.kp = 0.001*(float)SPID.OP;
280 //         ultra_pid.ki = 0.001*(float)SPID.OI;
281 //         ultra_pid.kd = 0.001*(float)SPID.OD;
282 //         wz_speed_pid.kp = 0.001*(float)SPID.IP;
283 //         wz_speed_pid.ki = 0.001*(float)SPID.II;
284 //         wz_speed_pid.kd = 0.001*(float)SPID.ID;
285 //     }
286 //     eso_att_inner_c[THRr].b0= SPID.YD;
287 // }
288 // else
289 // {
290 //     ultra_pid_safe.kp = 0.001*(float)SPID.OP;
291 //     ultra_pid_safe.ki = 0.001*(float)SPID.OI;
292 //     ultra_pid_safe.kd = 0.001*(float)SPID.OD;
293 //     wz_speed_pid_safe.kp = 0.001*(float)SPID.IP;
294 //     wz_speed_pid_safe.ki = 0.001*(float)SPID.II;
295 //     wz_speed_pid_safe.kd = 0.001*(float)SPID.ID;
296 // }
297 }////nav 01

```

可以修改各 SPID 参数被那个飞控参数使用，用户可自行修改。源码默认提供了姿态参数调整模式，通过将 NRF-G 模块的 8 个红色拨码开关都扒到上进入该模式，模式判断实际是在 ucos.c 中

```

263 //
264 mode.att_pid_tune=KEY[6]&&KEY[5]&&KEY[3]&&KEY[2]&&KEY[1]&&KEY[0];
265 //

```

处，进入该模式后飞控只控制单轴，且定高模式为油门直接输出模式，调参时将飞机通过烤 x 型四轴的方式固定，解锁飞行器，推动油门（油门会被 att_tuning_thr_limit 限制该参数最好设为起飞油门），通过观察飞行器姿态进行参数调整，NRF-G 模块第二个两档开关可以触发 15° 的姿态阶跃通过不断给定阶跃的方式整定参数，PC 上位机在该模式只会显示期望角度和设定角度用户可以通过曲线观察姿态情况，各参数对应情况在 rc_mine.c 文件中：

```

220 //
227 ctrl_2.PID[PIDPITCH].kp = ctrl_2.PID[PIDROLL].kp = 0.001*SPID.OP;
228 ctrl_2.PID[PIDPITCH].ki = ctrl_2.PID[PIDROLL].ki = 0.001*SPID.OI;
229 ctrl_2.PID[PIDPITCH].kd = ctrl_2.PID[PIDROLL].kd = 0.001*SPID.OD;
230 ctrl_1.PID[PIDPITCH].kp = ctrl_1.PID[PIDROLL].kp = 0.001*SPID.IP;
231 ctrl_1.PID[PIDPITCH].ki = ctrl_1.PID[PIDROLL].ki = 0.001*SPID.II;
232 ctrl_1.PID[PIDPITCH].kd = ctrl_1.PID[PIDROLL].kd = 0.001*SPID.ID;
233 // ctrl_2.PID[PIDYAW].kp = 0.001*SPID.YP;
234 // ctrl_2.PID[PIDYAW].ki = 0.001*SPID.YI;
235 // ctrl_2.PID[PIDYAW].kd = 0.001*SPID.YD;
236 //
237 eso_att_outter_c[PITr].b0=eso_att_outter_c[ROLr].b0= SPID.YD;
238 if(SPID.YP!=0)
239 att_tuning_thr_limit=SPID.YP;

```

其中要注意的时 SPID.YD 对应的是自抗扰控制的关键参数调参时最好先设为 0，整定内外环 PID 当达到不震荡能快速逼近期望但是存在残差时再调整 100 开始调整 b0 如果出现超调则增大，降低 b0 能获得更快的响应速度但是会出现抖动和震荡，YP 对应油门限制，当参数调节完成后将获得的参数（注意使用的是*0.001 后的 APP 设定值）在 Parmameter.c 中进行修改重新烧写估计。

对于高度和定点 PID 用户可通过修改代码对 pid.nav，ultra_pid 和 wz_speed_pid 进行调节。

3. 飞控试飞流程

- 1：组装好机架，将电机、电调安装到机架上，并连接好电调和电机，先不安装螺旋桨；
- 2：将飞控安装到机架上，并按照顺序，将 1234 号电调信号线插到飞控相应位置，完成电调校准；
- 3：将接收机按照接口定义，连接到飞控上，调整程序使各开关满足上节映射要求；
- 4：完成传感器校准，倾斜飞行器查看姿态角是否符合实际情况（其中_fc 的为 FC 模块解算结果，不带的为

IMU 模块解算结果)

▼ ftempreat...	0	float
Pit_fc	0	float
RoI_fc	0	float
Pitch	0	float
Roll	0	float
baroAlt_fc	0	int
MS5611_Press...	0	float
Yaw_fc	0	float
Yaw	0	float
...

5：插上飞机电池，使用遥控手动油门控制模式，解锁飞机，小幅度推动油门，此时各个电机开始旋转，观察电机旋转方向是否和手册定义相同，如果不同，交换电调和电机三根导线中任意两根导线的位置即可；

6：电机旋转时，分别推拉 ROL 和 PIT 摇杆，观察电机是否按照相应逻辑调整转速（向左推 rol 摇杆，右边两个电机加速，左边两个电机减速，向下拉 pit 摇杆，前方两个电机加速，后方两个电机减速）；

7：按照电机转向，安装相应螺旋桨，螺旋桨安装后，一定要用螺丝刀插入螺旋桨固定帽的小孔中，用力旋紧，防止飞行中螺旋桨飞出；

8：在空旷处，远离人群，使用气压定高和自稳模式进行试飞，无误后使用超声波光流飞行即可定点悬停。

4. 注意事项

4.1 定高

(1) 模式切换

定高模块切换没有条件限制可以在飞行中随意切换但是最好不要切换到手动油门模式各模式说明如下：

手动油门模式（height_ctrl_mode==0）：

该模式遥感直接控制油门，遥感低则油门为 0，高为油门最高

气压计定高模式（height_ctrl_mode==1）

该模式使用气压计高度作为外环，气压计加速度计融合速度为内环，遥感控制对应垂直方向速度量，中位则保持当前高度

超声波定高模式（height_ctrl_mode==2）

该模式使用超声波高度作为外环，为方便调制该模式默认定高为固定值 1.2m 用户如果不想使用这种方法可以将 height_ctrl.c 中：

```
586     if(height_ctrl_mode_use==2&&!mode.height_safe&&ALT_POS_SONAR2>0.35)  
587         exp_height=1200;  
588     // if(EXP_Z_SPEED==0)
```

两句话注释掉，气压计加速度计融合速度为内环，遥感控制对应垂直方向速度量，中位则保持当前设定高度 1.2m

(2) 注意事项

1：定高模式相当于让飞机自己控制油门，危险性高，请大家使用定高模式飞行时，一定要注意安全，附近不要有人，飞机也要做好防撞措施，请有飞行经验的操作手进行试飞，因开源程序，我们不对任何飞行造成的损失负责。

2：试飞前务必对加速度计进行校准，因为定高需要依靠惯导，加速度计不经校准，严重影响定高控制。

3：切换不同定高模式，推荐在飞机已经降落，锁定的状态下进行切换，如果在控制进行切换时切换瞬间会将当前高度设为期望值，如果是切入超声波模式则高度设置值固定为 1.2m。

4：因为定高采用了惯导，在螺旋桨动平衡不佳，振动较大的飞机上，可能出现异常，无法正常定高。

5：使用气压或者超声波定高模式起飞和飞行时，请缓慢推动油门，让飞机缓慢上升下降。

6：超声波容易受到螺旋桨气流的影响，所以超声波安装位置最好选择飞机上收气流影响比较小的位置，如飞机中心板附近。

飞行注意事项：

1：定高模式下，50%油门为保持当前飞行高度，低于 50%下降，高于 50%上升。

2：用气压计模式飞行时，由于天气影响、气流影响等原因，飞机会缓慢上升下降。

再次提醒：注意定高模式下的飞行安全，人身安全和飞机安全，做好防护措施。

4.2 定点模式

(1) 模式说明

定点模式目前使用光流模块作为主要传感器，主要分为手动模式和光流定点模式：

手动自稳模式 (`mode.flow_hold_position==0`)：

该模式遥杆直接控制飞行器期望姿态，悬停时飞行器位置会发生飘移且没有遥感回中的制动过程

光流定点模式 (`mode.flow_hold_position==1`)：

该模式遥感控制的是机体 xy 方向的移动速度，遥感中位时悬停在当前位置，飞行时遥感回中会快速制动

(2) 注意事项

1：定高模式相当于让飞机自己期望姿态，危险性高，请大家使用光流模式飞行时，一定要注意安全，附近不要有人，飞机也要做好防撞措施，请有飞行经验的操作手进行试飞，因开源程序，我们不对任何飞行造成的损失负责。

2：试飞前务必对加速度计进行校准，因为光流 UKF 滤波需要依靠惯导，加速度计不经校准，严重影响定点控制。

4：因为 UKF 采用了惯导，在螺旋桨动平衡不佳，振动较大的飞机上，可能出现异常，无法正常定点。

5：使用光流模式飞行时由于受光照和地面纹理条件影响不能保证都有良好的定点效果和快速制动，因此侧飞时请缓慢推动遥感。

6：光流数据融合使用了超声波高度信息其容易受到螺旋桨气流的影响，所以超声波安装位置最好选择飞机上收气流影响比较小的位置，如飞机中心板附近。

飞行注意事项：

1：注意光流模块的安装方向：



镜头朝向飞机尾部，注意一定不要安装错误，安装错误造成的飞行器炸鸡概不负责。

再次提醒：注意定点模式下的飞行安全，人身安全和飞机安全，做好防护措施。

5. 进一步开发说明

由于 OLD-X 飞控中包括了很多实验室保密内容因此属于半开源性质，如 IMU 模块中的内容只提供可下载的更新固件不提供源码和原理图，而 FC，NRF，NRF-G 模块都提供源码和原理图，当然 IMU 模块中采集的传感器数据是会发送到 FC 模块中用户可以自行对数据进行处理，相比其他开源飞控 OLD-X 虽然也移植了很多国内外优秀开源飞控的内容（如 PX4，Autoquad，匿名等）但相比照搬现有飞控只重新做了 PCB 的开源飞控来说，我们增加了自抗扰控制算法，提供了 EKF 气压计的 lib，使用了 UcosII 操作系统未来还将自己 H-inf 姿态控制，自适应控制，UKF 的 GPS 数据融合等程序都已在源码构建测试，这些都是其他飞控难以媲美的，有可能未来我们的程序也会被其他开源飞控移植过去但是由于有 IMU 闭源部分的更新任然能保证我们飞控在现有飞控中数据融合的高性能和飞行稳定性，OLD-X 飞控最终是希望能够像 DJI M100 一样只使用 SDK 进行无人机的控制，因为目前飞控研究在国内外已经趋于冷淡，我们下一步更加注重于视觉智能控制在无人机上的应用，相信 OLD-X 飞控只是一个研发框架和平台，希望用户能在上面快速方便的开发出自己的无人机创意项目。